BUILD PROJECTS · BUILDER · 20 MIN READ

Project: build a club sign-up app

A guided build with copy-ready prompts: a public sign-up form, a private organiser list, safe database rules, and a live link to share.

Pass the check (4 of 5) online to earn the “Club Builder” badge:
https://digitalbuilders757.com/university/study/project-club-sign-up-app

What you'll build

A real tool for a club, team, class or community group: anyone can sign up with a short form, and only the organiser can sign in to see the list and download it. It's small enough for an afternoon and teaches the pieces almost every app needs: a form, a database, sign-in, permissions and deployment.

Step 1: Plan it in ten minutes

Write three user stories with 'done when' lines. For example: 'As a visitor, I want to sign up in under a minute, so I don't give up.' Done when: the form has name, email and one interest, and shows a thank-you message. 'As the organiser, I want to see and download sign-ups.' Done when: only I can open the list, and it exports a CSV. Out of scope for version one: payments, emails and member profiles.

Step 2: Build the screens

Prompt your app builder: 'Build a club sign-up app with two pages. A public page with the club name, a one-line description and a form (name, email, interest from a dropdown) that shows a friendly thank-you after submitting. An organiser page that requires sign-in and lists sign-ups newest first with a Download CSV button. Mobile-first, clear labels, accessible.' Then click through both pages on your phone before adding anything else.

Step 3: Save data safely

Connect a database (for example Supabase) and ask: 'Store sign-ups in a table with row level security on. Anyone may insert a sign-up; only the signed-in organiser may read them. Explain each policy in plain English.' Test it like an attacker: sign out and try to open the organiser page, and check that a visitor can't read other people's details. Keep keys in environment variables, never in the code.

Step 4: Ship, share and improve

- Save your work to GitHub and deploy it to get a live link (for example with Vercel).

- Run the launch checklist: a test sign-up from a phone, the organiser view, the CSV and the thank-you message.
- Share it with your real club and ask two people what confused them.
- Improve one thing from their feedback, and mark the Club sign-up Build Challenge as built.

Key terms

User story
A need written as: As a [user], I want [goal], so that [reason].

Acceptance criteria
Testable conditions that say when a user story is done.

Database
Organised storage for everything an app must remember.

Row level security (RLS)
Database rules that decide which rows each user may read or change.

Authentication
Checking who someone is (signing in).

Environment variable
A setting or secret the app reads at run time, kept outside the code.

Deploy
Publish a version of your app to a host.

CSV
A plain-text table with values separated by commas, for moving data between tools.

Try it: Build it this week

1. Write the three user stories and the out-of-scope list.
2. Use the Step 2 prompt to build both pages, and check them on your phone.
3. Add the database with row level security and test it signed out.
4. Deploy, share the link with your club and fix one thing from their feedback.

KNOWLEDGE CHECK

Project: build a club sign-up app

5 questions. Circle one answer for each; 4 or more correct passes. Answers are at the back.

1. Who should be able to read the list of sign-ups?

A. Anyone with the link
B. Every person who signed up, so they can see who's coming
C. Only the signed-in organiser
D. Search engines

2. Which database rule setup matches this app?

A. RLS off so the form works
B. RLS on with no policies at all
C. Put the service role key in the form
D. RLS on: anyone may insert a sign-up; only the organiser may select

3. Best way to check your security works?

A. Sign out and try to open the organiser page
B. Trust that the AI set it up, since you asked it to make it secure
C. Ask a friend who codes to look at the page and say whether it looks secure
D. Wait for a user to report a leak

4. Which belongs in 'out of scope' for version one?

A. The sign-up form
B. Payments and automatic emails
C. The organiser's list
D. Working on phones as well as laptops

5. After sharing it with your club, what's the most useful next step?

A. Add ten new features at once so the club has more reasons to use it
B. Rebuild it in a different tool that's more popular with clubs
C. Ask two people what confused them and fix one thing
D. Hide the link until every page looks perfect on every device

Official sites

Supabase docs: Row Level Security: https://supabase.com/docs/guides/database/postgres/row-level-security
Vercel docs: Environments (preview and production): https://vercel.com/docs/deployments/environments

Last reviewed 2026-11-25. AI tools change fast; check each tool's own site for current features, plans and prices.

ANSWER KEY

Project: build a club sign-up app

1. C: Only the signed-in organiser
Sign-ups are personal data. Visitors may add themselves, but only the organiser reads the list.

2. D: RLS on: anyone may insert a sign-up; only the organiser may select
Narrow policies for each action keep the form open and the list private.

3. A: Sign out and try to open the organiser page
Test like an attacker: as a signed-out visitor, you should be blocked.

4. B: Payments and automatic emails
Ship the core first; extras come after real people use it.

5. C: Ask two people what confused them and fix one thing
Small improvements from real feedback beat guessing.